Knapsack problem with objective value gaps

Alexandre Dolgui, Mikhail Y. Kovalyov, Alain Quilliot

Knapsack problem with objective value gaps

Alexander Dolgui[a], Mikhail Y. Kovalyov[b,*], Alain Quilliot[c]

[a] *Ecole des Mines de Nantes, CNRS UMR 6597 IRCCYN, La Chantrerie, 4, rue Alfred Kastler - B.P. 20722, F-44307 Nantes Cedex 3, France, E-mail: alexandre.dolgui@mines-nantes.fr*

[b] *United Institute of Informatics Problems, National Academy of Sciences of Belarus, Minsk, Belarus, E-mail: kovalyov_my@newman.bas-net.by*

[c] *LIMOS, UMR CNRS 6158, Bat. ISIMA, Université Blaise Pascal, Campus des Cézeaux, BP 125, 63173 Aubiere, France, E-mail: alain.quilliot@isima.fr*

Abstract

We study a 0-1 knapsack problem, in which the objective value is forbidden to take some values. We call gaps related forbidden intervals. The problem is NP-hard and pseudo-polynomially solvable independently on the measure of gaps. If the gaps are large, then the problem is polynomially non-approximable. A non-trivial special case with respect to the approximate solution appears when the gaps are small and polynomially close to zero. For this case, two fully polynomial time approximation schemes are proposed. The results can be extended for the constrained longest path problem and other combinatorial problems.

Keywords: knapsack problem; computational complexity; fully polynomial time approximation scheme; forbidden values

1 Introduction

There exist practical optimization problems, in which some of the objective function values are forbidden. We study computational complexity and approximation issues of the following 0-1 knapsack type problem, which we denote K-Gaps. Let us consider non-negative integer numbers a_j, b_j, $j = 1, \ldots, n$, $\underline{f}_i$, $\overline{f}_i$, $i = 1, \ldots, k$, and B, where $\underline{f}_1 \le \overline{f}_1 < \underline{f}_2 \le \overline{f}_2 < \cdots < \underline{f}_k \le \overline{f}_k \le A$ and $A = \sum_{j=1}^{n} a_j$. Denote $x = (x_1, \ldots, x_n)$ and $[\underline{f}, \overline{f}] = \{\underline{f}, \underline{f}+1, \ldots, \overline{f}\}$.

Problem K-Gaps:

$$\max F(x) = \sum_{j=1}^{n} a_j x_j, \text{ subject to}$$

*Corresponding author

$$G(x) = \sum_{j=1}^{n} b_j x_j \leq B,$$

$$F(x) \notin \{[\underline{f}_1, \overline{f}_1], \ldots, [\underline{f}_k, \overline{f}_k]\},$$

$$x_j \in \{0, 1\}, \quad j = 1, \ldots, n.$$

Denote optimal objective function value of this problem as F^*. We call intervals $[\underline{f}_i, \overline{f}_i]$, $i = 1, \ldots, k$, *objective value gaps*, or simply, gaps. We call intervals in $[0, A]$ other than gaps *feasible intervals*.

Problem K-Gaps is a generalization of the classic 0-1 knapsack problem which we denote as Knapsack. We observed problem K-Gaps in the following two practical situations.

In the first situation, there is a logistics provider which consolidates orders of various customers and delivers them by a fleet of vehicles of different weight capacities, for example, by 3 vehicles of 1 tonne, 2 vehicles of 3 tonnes and 2 vehicles of 20 tonnes. Each vehicle must be loaded to its full capacity and it can make at most one trip in the same day. The provider can deliver at most 45 tonnes of all orders in a particular day, which is its loading capacity bound in this day. There are n orders. If an order is accepted for delivery, then it must be satisfied in full but it can be split between the vehicles. Each order j is associated with a weight a_j in tonnes and a value b_j of a single non-renewable resource to be consumed. The total resource consumption is limited by B. Define $x_j = 1$ if order j is selected for delivery and $x_j = 0$, otherwise. The problem is to select a subset of orders such that the total weight $F(x) = \sum_{j=1}^{n} a_j x_j$ is maximized, subject to $\sum_{j=1}^{n} b_j x_j \leq B$, $F(x) \notin \{[10, 19], [30, 39]\}$ and $x_j \in \{0, 1\}$, $j = 1, \ldots, n$. The gaps contain total weights that cannot be realized by the available vehicles if they are loaded to their full capacities.

In the second situation, a company can select a subset of n orders to be processed in a specific day. Each selected order j is associated with a profit a_j and a capacity consumption b_j, both are positive integer numbers. The total capacity is bounded by B. If the total profit is less than a given value $V > 0$, then it is not worth doing something in this day and it is better to make this day the day off. Define $x_j = 1$ if order j is accepted for processing and $x_j = 0$ otherwise. The problem is to select a subset of orders such that $F(x) = \sum_{j=1}^{n} a_j x_j$ is maximized, provided that $\sum_{j=1}^{n} b_j x_j \leq B$, $F(x) \notin [0, V - 1]$, and $x_j \in \{0, 1\}$, $j = 1, \ldots, n$. If an optimal solution of this problem cannot be found within the required time or memory limit and an approximate solution is the goal, then avoiding profit values smaller than V is an issue.

Good sources of information on knapsack type problems can be found in monographs of Martello and Toth [11] and Kellerer, Pferschy and Pisinger [8]. We are not aware of any results on the knapsack problems or any other optimization problems with the objective values gaps. In the next section, we comment on the computational complexity of the exact and approximate solutions of the problem K-Gaps and its special cases. Section 3 studies the case with gaps close to zero. The paper concludes with a short summary of the results and a discussion of their extension for other optimization problems.

2 Computational complexity

Let P_L be a polynomial function of the input length L of the problem K-Gaps in binary encoding, $L = O\Big(\log_2\Big(B + \sum_{j=1}^{n}(a_j + b_j) + \sum_{i=1}^{k}(\overline{f}_i + \underline{f}_i)\Big)\Big)$. We denote by K-Small-Gaps and K-Large-Gaps special cases of the problem K-Gaps, in which the total measure of gaps does not exceed P_L and the total measure of feasible intervals does not exceed P_L, respectively. The following propositions can be easily proved.

Proposition 1 *Any of the problems* K-Small-Gaps *and* K-Large-Gaps *is NP-hard.*

An idea of the proof of this proposition for the problem K-Small-Gaps is that introducing small gaps, for example, one gap $[1, 1]$, to the classic problem Knapsack does not make it easier. An idea of the proof for the problem K-Large-Gaps is that a special case of this problem with $a_j = b_j$, $j = 1, \ldots, n$, and two gaps $[0, B-1]$ and $[B+1, 2B]$, where $\sum_{j=1}^{n} b_j = 2B$, is the well known NP-complete problem Partition.

Let us comment on the approximability of the problem K-Gaps. An approximation algorithm for the problem K-Gaps is called to have a *performance guarantee* Δ, $0 \leq \Delta \leq 1$, if, for any instance of this problem, it finds a feasible solution with the objective function value $F^0 \geq \Delta F^*$ in the case when a feasible solution exists. Such an algorithm is called a Δ-*approximation* algorithm and the solution it delivers is called a Δ-*approximate* solution. A *Fully Polynomial Time Approximation Scheme (FPTAS)* for the problem K-Gaps is a collection of algorithms, $\{H_\varepsilon\}$, such that, for any given ε, $0 < \varepsilon \leq 1$, and any instance of the problem K-Gaps, algorithm H_ε finds a feasible solution with objective function value $F^0 \geq (1-\varepsilon)F^*$, if a feasible solution exists, and it does it in time polynomially bounded by the instance length in binary encoding and $1/\varepsilon$. FPTASes are a popular solution tool for NP-hard problems, see, e.g., recent papers of Schemeleva et al. [18], Li and Li [10], Halman et al. [3], Nguyen et al. [13] and Samanta et al. [17].

Note that the classic problem KNAPSACK, while NP-hard, admits very efficient approximation algorithms such as the folkloric $O(n \log n)$ time greedy algorithm that guarantees $\Delta = 1/2$ and several FPTASes (cf. Martello and Toth [11] and Kellerer and Pferschy [7]). However, problem K-GAPS is hard to approximate.

Proposition 2 *No polynomial time approximation algorithm with any performance guarantee, including FPTAS, exists for the problem* K-LARGE-GAPS*, unless* $\mathcal{P} = \mathcal{NP}$.

An idea of the proof of this proposition is that a polynomial time approximation algorithm with any performance guarantee for the problem K-LARGE-GAPS with $a_j = b_j$, $j = 1, \ldots, n$, and two gaps $[0, B-1]$ and $[B+1, 2B]$, $\sum_{j=1}^{n} b_j = 2B$, would be a polynomial algorithm for the NP-complete problem PARTITION, which is impossible, unless $\mathcal{P} = \mathcal{NP}$.

Note that the existence of an FPTAS or a polynomial time approximation algorithm for the problem K-SMALL-GAPS is an open question.

Recall that $A = \sum_{j=1}^{n} a_j$. Problem K-GAPS can be solved in $O(nA)$ time by a modification of the well known *profit based* dynamic programming algorithm (cf. Kellerer and Pferschy [7]) for the classic problem KNAPSACK. The modification is to exclude forbidden profit values from the last table of state variables. Thus, problem K-GAPS is NP-hard and pseudo-polynomially solvable.

3 Problem with gaps polynomially close to zero

Consider a special case of the problem K-GAPS, in which the largest gap value is polynomially bounded: $\overline{f}_k \leq P_L$. We denote this special case as K-LEFT-SMALL-GAPS. Note that the problem K-LEFT-SMALL-GAPS is NP-hard.

3.1 Reduction to a series of classic 0-1 knapsack problems

Consider the problem K-LEFT-SMALL-GAPS. Let $a_1 \leq \cdots \leq a_n$. There are three cases to analyze: (1) $\overline{f}_k < a_1$, (2) $a_n \leq \overline{f}_k$, and (3) $a_1 \leq \cdots \leq a_t \leq \overline{f}_k < a_{t+1} \leq \cdots \leq a_n$, $1 \leq t \leq n-1$.

Case (1) has two sub-cases. Sub-case (1.1): zero does not belong to a gap, that is, $\underline{f}_1 > 0$, and sub-case (1.2): zero belongs to a gap, that is, $\underline{f}_1 = 0$. Let D_K and D_G denote feasible domains of the problems KNAPSACK and K-LEFT-SMALL-GAPS, respectively. In the sub-case (1.1), $D_G = D_K$. Therefore, the two problems are equivalent in this sub-case.

Denote n-dimensional vector $e_0 = (0, \ldots, 0)$. In the sub-case (1.2), $D_G = D_K \setminus \{e_0\}$. Therefore, any algorithm for the problem KNAPSACK, which delivers a non-zero solution of the required quality, is the algorithm for the problem K-LEFT-SMALL-GAPS of the same quality. Many algorithms for the problem KNAPSACK, including dynamic programming algorithms, greedy 1/2-approximation algorithm and FPTASes described in Martello and Toth [11] and Kellerer, Pferschy and Pisinger [8], satisfy this property. Therefore, they can be employed to solve the sub-case (1.2).

In the case (2), all profit coefficients are bounded from above by $\overline{f}_k \leq P_L$. Therefore, the standard profit based dynamic programming algorithm with the running time $O(nA) \leq O(n^2 P_L)$ can be used to find a feasible solution, $x^{(f)}$, for each possible objective function value $f \in \{0, 1, \ldots, nP_L\}$ in the problem KNAPSACK such that $F(x^{(f)}) = f$. We can set $F(x^{(f)}) = -\infty$ if a feasible solution with value f does not exist. Solution of the problem K-LEFT-SMALL-GAPS in the case (2), x^*, can be found from

$$F(x^*) = \max\Big\{F(x^{(f)}) \mid f \in \{0, 1, \ldots, nP_L\}, f \notin [\underline{f}_1, \overline{f}_1] \cup \cdots \cup [\underline{f}_k, \overline{f}_k]\Big\}.$$

Thus, the problem K-LEFT-SMALL-GAPS is solvable in $O(n^2 P_L)$ time in the case (2).

Consider case (3). Observe that only selection of the numbers from $a_1, \ldots, a_t$ can lead to an objective value gap and, if at least one number from $a_{t+1}, \ldots, a_n$ is selected, then the objective function value does not fall into a gap. Let us solve problem KNAPSACK to optimality. Denote by x^0 an optimal solution of this problem. If $x^0_j = 1$ for $j \geq t+1$, then x^0 is also optimal for the problem K-LEFT-SMALL-GAPS in the case (3). If $x^0_j = 0$ for $j = t+1, t+2, \ldots, n$, then we know that $\sum_{j=1}^{t} a_j \leq tP_L$ is an upper bound on the optimal objective function value in both problems KNAPSACK and K-LEFT-SMALL-GAPS. In the latter case, both problems can be solved in $O(t^2 P_L)$ time by the standard profit based dynamic programming algorithm. Thus, the problem K-LEFT-SMALL-GAPS can be solved to optimality in $O(T + t^2 P_L)$ time in the case (3), where $O(T)$ is the time of solving the problem KNAPSACK to optimality.

Assume that an optimal algorithm for the problem KNAPSACK is not appropriate because of its time or space requirements. Then the following approach for the problem K-LEFT-SMALL-GAPS in the case (3) can be applied. Consider a *reduced* problem KNAPSACK, in which variables are limited to $x_1, \ldots, x_t$. Apply the standard profit based dynamic programming algorithm to this problem and find a feasible solution, $x^{(f)} = (x^{(f)}_1, x^{(f)}_2, \ldots, x^{(f)}_t)$, for each possible objective function value $f \in \{0, 1, \ldots, tP_L\}$ such that $F(x^{(f)}) = f$ and $G(x)$

is minimized on the set of solutions satisfying $F(x) = f$. Set $F(x^{(f)}) = -\infty$ if a feasible solution with value f does not exist. The run time of this algorithm will be $O(t^2 P_L)$ in this case. Define $F_f = F(x^{(f)})$ and $G_f = G(x^{(f)})$, $f = 0, 1, \ldots, tP_L$. For each pair (F_f, G_f), $F_f \neq -\infty$, formulate the following problem denoted as f-KNAPSACK.

Problem f-KNAPSACK:

$$\max \sum_{j=t+1}^{n} a_j x_j, \text{ subject to}$$

$$\sum_{j=1}^{n} b_j x_j \leq B - G_f,$$

$$x_j \in \{0, 1\}, \quad j = t+1, t+2, \ldots, n.$$

Apply a Δ-approximation algorithm to solve problems f-KNAPSACK for $f = 0, 1, \ldots, tP_L$. Let $\bar{x}^{(f)}$ denote the Δ-approximate solution obtained. It can be easily verified that a solution $(x^{(f)}, \bar{x}^{(f)}) = (x_1^{(f)}, \ldots, x_t^{(f)}, \bar{x}_{t+1}^{(f)}, \ldots, \bar{x}_n^{(f)})$ such that $F(x^{(f)}, \bar{x}^{(f)}) = \sum_{j=1}^{t} a_j x_j^{(f)} + \sum_{j=t+1}^{n} a_j \bar{x}_j^{(f)}$ is maximized, provided that $F(x^{(f)}, \bar{x}^{(f)}) \notin [\underline{f}_1, \overline{f}_1] \cup \cdots \cup [\underline{f}_k, \overline{f}_k]$, over values f such that $f = 0, 1, \ldots, tP_L$, $F_f \neq -\infty$, is the Δ-approximate solution for the problem K-LEFT-SMALL-GAPS in the case (3). It can be found in $O(t^2 P_L + tP_L T_{n-t}^{(\Delta)})$ time, where $O(T_{n-t}^{(\Delta)})$ is the running time of a Δ-approximation algorithm for the problem KNAPSACK with $n - t$ variables.

The results of this sub-section are summarized in the following statement.

Statement 1 *An optimal solution of the problem* K-LEFT-SMALL-GAPS *can be found in* $O(T + t^2 P_L)$ *time and a* Δ*-approximate solution of this problem can be found in* $O(t^2 P_L + tP_L T_{n-t}^{(\Delta)})$ *time.*

3.2 Two FPTASes for the problem K-LEFT-SMALL-GAPS

There exist FPTASes for the problem KNAPSACK, for example, an FPTAS which runs in $O(n^2/\varepsilon)$ time, see Sahni and Horowitz [16]. With this FPTAS, our Δ-approximation approach in the previous sub-section provides FPTAS for the problem K-LEFT-SMALL-GAPS with $O(t^2 P_L + tP_L(n-t)^2/\varepsilon)$ running time. Below we will describe a different FPTAS, denoted as $\{H_\varepsilon\}$, for this problem.

Algorithm H_ε is a modification of the *interval partitioning* technique of Sahni [15] and it differs from this technique in that the measure of closeness of partial solutions with respect to the objective function value is different in the intervals $[0, \overline{f}_k]$ and $[\overline{f}_k + 1, F^{(j)}]$, where $F^{(j)}$

is the maximum objective function value in iteration j. The measure of closeness is equal to zero in the former interval and it is equal to a certain value $\delta_j > 0$ in the latter interval. Denote n-dimensional vector $e_j = (0, \ldots, 0, 1, 0, \ldots, 0)$, where the unit is in the position j, $j = 1, \ldots, n$.

Algorithm H_ε (FPTAS for the problem K-Left-Small-Gaps)

Step 1 (Initialization) Set $X_0 = \{e_0\}$, $F(e_0) = G(e_0) = 0$ and $j = 1$.

Step 2 (Recursion) Calculate set $Y_j = \{x, x + e_j \mid x \in X_{j-1}\}$. Compute $F(x)$ and $G(x)$ for each $x \in Y_j$. Calculate set $Z_j = \{x \in Y_j \mid G(x) \leq B\}$. If $j = n$, then perform Step 3. Otherwise, perform procedure *Partition*(Z_j) given below to partition set Z_j into disjoint subsets $Z_{j1}, Z_{j2}, \ldots, Z_{jk_j}$. In each subset Z_{ji}, select vector $x^{(ji)}$ which minimizes $G(x)$, $i = 1, \ldots, k_j$. Calculate $X_j = \{x^{(j1)}, x^{(j2)}, \ldots, x^{(jk_j)}\}$. Re-set $j := j+1$ and repeat Step 2.

Step 3 (Approximate solution) Select an approximate solution $x^{(\varepsilon)}$ such that

$$F(x^{(\varepsilon)}) = \max\{F(x) \mid x \in Z_n, F(x) \notin [\underline{f}_1, \overline{f}_1] \cup \cdots \cup [\underline{f}_k, \overline{f}_k]\}.$$

Procedure *Partition*(Z_j)

Step 1 Re-number vectors of the set Z_j such that $F(x^{(1)}) \leq F(x^{(2)}) \leq \cdots \leq F(x^{(r_j)}) \leq \overline{f}_k < F(x^{(r_j+1)}) \leq \cdots \leq F(x^{(h_j)})$. Let there be r_j^* distinct values among $F(x^{(1)}), F(x^{(2)}), \ldots, F(x^{(r_j)})$. It is clear that $r_j^* \leq P_L$.

Step 2 Assign vectors $x^{(1)}, x^{(2)}, \ldots, x^{(r_j)}$ with the same value $F(x)$ to the same subset Z_{ji}, $i = 1, \ldots, r_j^*$.

Step 3 Compute $\delta_j = \varepsilon F(x^{(h_j)})/n$. Assign vectors $x^{(r_j+1)}, x^{(r_j+2)}, \ldots, x^{(h_j)}$ with the same value $\lfloor F(x)/\delta_j \rfloor$ to the same subset Z_{ji}, $i = r_j^* + 1, r_j^* + 2, \ldots, k_j$. It can be easily seen that $k_j \leq P_L + F(x^{(h_j)})/\delta_j = P_L + n/\varepsilon$.

The time and space requirements of the algorithm H_ε can be evaluated as $O(nP_L + n^2/\varepsilon)$.

Theorem 1 *The family of algorithms* $\{H_\varepsilon\}$ *constitutes an FPTAS for the problem* K-Left-Small-Gaps.

Proof. Since the running time of the algorithm H_ε satisfies the definition of an FPTAS, we only need to prove that relation $F(x^{(\varepsilon)}) \geq (1-\varepsilon)F^*$ is satisfied. Consider an optimal solution $x^* = (v^*, u^*) = (v_1^*, \ldots, v_j^*, u_{j+1}^*, \ldots, u_n^*)$, $1 \leq j \leq n-1$. Denote $x^{(v^*)} = (v^*, 0, \ldots, 0)$ and assume without loss of generality that $x^{(v^*)} \in Z_{ji}$, $1 \leq i \leq k_j$. If $i \leq r_j^*$, then, for the only partial solution $x^{(ji)}$ selected from the set Z_{ji}, we have $F(x^{(ji)}) = F(x^{(v^*)})$. If $i > r_j^*$, then $F(x^{(v^*)}) > \overline{f}_k$, $F(x^{(ji)}) > \overline{f}_k$ and $|F(x^{(ji)}) - F(x^{(v^*)})| \leq \delta_j \leq \varepsilon F^*/n$. In both cases, $G(x^{(ji)}) \leq G(x^{(v^*)})$. Hence, complete solution $x' = x^{(ji)} + (0, \ldots, 0, u^*)$, extended from the partial solution $x^{(ji)}$ in the same way as the optimal solution x^* is extended from $x^{(v^*)}$, is feasible and $|F(x') - F(x^*)| \leq \varepsilon F^*/n$. Now let $x' = (v', u')$, where $x^{(v')} := (v', 0, \ldots, 0) \in Z_{j'i'}$, $j < j' \leq n-1$, $1 \leq i' \leq k_{j'}$. Similarly, we can show that $x'' = x^{(j'i')} + (0, \ldots, 0, u')$ is feasible and $|F(x'') - F(x')| \leq \varepsilon F^*/n$. From $|F(x') - F(x^*)| \leq \varepsilon F^*/n$ and $|F(x'') - F(x')| \leq \varepsilon F^*/n$, we deduce that $|F(x'') - F(x^*)| \leq 2\varepsilon F^*/n$. Continuing in the same fashion, we can show that there exists vector $x^{(\varepsilon)} \in Z_n$ which is feasible and satisfies $|F(x^{(\varepsilon)}) - F(x^*)| \leq \varepsilon F^*$. The latter relation implies $F(x^{(\varepsilon)}) \geq (1-\varepsilon)F^*$, as it is required. ∎

The results of this sub-section are summarized in the following statement.

Statement 2 *There exist FPTASes for the problem* K-Left-Small-Gaps *with the running times* $O(t^2 P_L + t P_L (n-t)^2/\varepsilon)$ *and* $O(nP_L + n^2/\varepsilon)$.

Note that the FPTASes and other algorithms in this section can be employed for fast solution of the second practical problem in Section 1 if $V - 1 \leq P_L$, that is, if the threshold on the lowest profit is polynomially bounded.

4 Conclusions and extensions

The problems K-Small-Gaps and K-Large-Gaps are NP-hard and pseudo-polynomially solvable, and the problem K-Large-Gaps cannot be approximated in polynomial time with any performance guarantee, unless $\mathcal{P} = \mathcal{NP}$. The problem K-Left-Small-Gaps is NP-hard, and FPTAS for the classic knapsack problem cannot be directly used for it. We suggested two FPTASes for this problem. Existence of an FPTAS or a polynomial time approximation algorithm for the more general problem K-Small-Gaps in which gaps are not necessarily close to zero, is an interesting open question.

The obtained results can be easily adapted for the constrained longest path problem, in which a simple path between two specified vertices of a directed graph has to be found such

that its total length is maximized, provided that the total value of the second parameter associated with the edges does not exceed a given threshold and the path length does not fall into the given gaps. The computational complexity propositions in Section 2 translate one-to-one for the constrained longest path problem with the objective value gaps. The exact algorithm and the FPTASes in Section 3 can be easily modified to solve this problem by addressing the specificity of this problem in a way similar to the dynamic programming algorithm and the FPTAS for the constrained shortest path problem with no gap, see Joksch [5] and Hassin [4], respectively.

There exist results for the *exact* combinatorial problems, which ask for the existence of a combinatorial structure of a given cost. Papadimitriou and Yannakakis [14] proved that the exact assignment problem is NP-complete in the ordinary sense, and Karzanov [6] developed polynomial time algorithm for the case of assignment costs limited to 0 and 1. Computational complexity of the exact assignment problem in unary encoding is an open question. Leclerc [9] and Barahona and Pulleyblank [2] suggested pseudo-polynomial time algorithms for the exact spanning tree problem, the exact perfect matching problem on planar graph, the exact cycle sum problem on planar directed graph and the exact cut problem on planar or toroidal graph. Milanic and Monnot [12] proved that the exact weighted independent set problem and the exact weighted maximum independent set problem are strongly NP-complete for cubic bipartite graphs and that these problems are pseudo-polynomial solvable for mK_2-free graphs, k-thin graphs (including interval graphs), chordal graphs, AT-free graphs, (claw,net)-free graphs, distance-hereditary graphs, circle graphs, graphs of bounded treewidth, graphs of bounded clique-width, and certain sub-classes of P_5-free and fork-free graphs. The results of Milanic and Monnot [12] imply that the exact perfect matching problem is pseudo-polynomially solvable for graphs with treewidth bounded by a constant. Vassilevska and Williams [19] and Abboud and Lewi [1] describe computational complexity results for exact weight subgraph problems, in which the number of nodes of the subgraph is a constant.

Observe that an exact discrete problem with value V and its counterpart with two gaps $[L, V-1]$ and $[V+1, U]$, where interval $[L, U]$ includes objective function values of all feasible solutions, are actually the same problem. Furthermore, the problem with gaps reduces to solving $U - L + 1$ respective exact problems with value $V = L, L+1, \ldots, U$. Therefore, the NP-completeness results for exact problems mentioned in the previous paragraph apply for

their counterparts with two gaps, the algorithm of Karzanov [6] can be employed to solve the assignment problem with gaps in polynomial time in the case of 0-1 costs, and the algorithms of Leclerc [9], Barahona and Pulleyblank [2] and Milanic and Monnot [12] can be employed to solve the relevant graph problems with gaps in pseudo-polynomial time. Computational complexity of the assignment problem with two gaps in unary encoding is an open question.

References

[1] Abboud, A., Lewi, K.: Exact weight subgraphs and the k-sum conjecture, In F.V. Fomin et al. (Eds.), ICALP 2013, Part I, LNCS 7965, pp. 1-12, Springer-Verlag Berlin Heidelberg, 2013.

[2] Barahona, F., Pulleyblank, W.R.: Exact aborescences, matchings and cycles. Discrete Applied Mathematics **16**, 91–99 (1987)

[3] Halman, N., Nannicini, G., Orlin, J.: A computationally efficient FPTAS for convex stochastic dynamic programs. SIAM J. Optim. **25**(1), 317–350 (2015)

[4] Hassin, R.: Approximation schemes for the restricted shortest path problem. Mathematics of Operations Research **17**, 36–42 (1992)

[5] Joksch, H.C.: The shortest route problem with constraints. Journal of Mathematical Analysis and Application **14**, 191–197 (1966)

[6] Karzanov, A.V.: Maximum matching of given weight in complete and complete bipartite graphs. Cybernetics **23**, 8–13 (1987) Translation from Kibernetika **1**, 7–11 (1987), in Russian.

[7] Kellerer, H., Pferschy, U.: A new fully polynomial time approximation scheme for the knapsack problem. Journal of Combinatorial Optimization **3**, 59–71 (1999)

[8] Kellerer, H., Pferschy, U., Pisinger, D.: Knapsack Problems, Springer-Verlag, Berlin, 2004.

[9] Leclerc, M.: Polynomial time algorithms for exact matching problems, Masters Thesis, University of Waterloo, Waterloo, 1986.

[10] Li, W., Li, J.: Approximation algorithms for k-partitioning problems with partition matroid constraint. Optimization Letters **8**(3), 1093–1099 (2014)

[11] Martello, S., Toth, P.: Knapsack Problems: Algorithms and Computer Implementations. Wiley, New York, 1990.

[12] Milanic, M., Monnot, J.: The exact weighted independent set problem in perfect graphs and related graph classes, Electronic Notes in Discrete Mathematics **35**, 317–322 (2009)

[13] Nguyen, T.H.C., Richard, P., Grolleau, E.: An FPTAS for response time analysis of fixed priority real-time tasks with resource augmentation. IEEE Transactions on Computers **64**(7), 1805–1818 (2015)

[14] Papadimitriou, C.H., Yannakakis, M.: The complexity of restricted spanning tree problems. Journal of the ACM **29**, 285–309 (1982)

[15] Sahni, S.: General techniques for combinatorial approximation. Operations Research **25**(6), 920–936 (1977)

[16] Sahni, S., Horowitz, E.: Combinatorial problems: reducibility and approximation. Operations Research **26**, 718–759 (1978)

[17] Samanta, R., Erzin, A.I., Raha, S., , Shamardin, Y.V., Takhonov, I.I., Zalyubovskiy, V.V.: A provably tight delay-driven concurrently congestion mitigating global routing algorithm. Applied Mathematics and Computation **255**, 92–104 (2015)

[18] Schemeleva, K., Delorme, X., Dolgui, A., Grimaud, F., Kovalyov, M.Y.: Lot-sizing on a single imperfect machine: ILP models and FPTAS extensions. Computers & Industrial Engineering **65**(4), 561–569 (2013)

[19] Vassilevska, V., Williams, R.: Finding, minimizing, and counting weighted subgraphs. In Mitzenmacher, M. (ed.), STOC, pp. 455-464, Association for Computing Machinery, 2009.